PubMed

December 1, 2021

1 Pubmed

[1]:
```
!export PATH=/Library/TeX/texbin:/Library/TeX/texbin/xelatex
import random
import math
import numpy as np
import pandas as pd
import time
```

1.0.1 Readfile Functions

[2]:
```
#input filename(str)
#output file(list)
def readfile(filename):
    with open(filename) as file_in:
        lines = []
        for line in file_in:
            lines.append(line)
    return lines
```

[3]:
```
import json
#input file(list)
#output json(list of dic)
def list2json(lines):
    jsons = []
    for i in range(len(lines)):
        tmp = json.loads(lines[i])
        jsons.append(tmp)
    return jsons
```

1.0.2 Calculus Functions

[4]:
```
# input: article
# output: number
def avg_token(article):
    if len(article)==0:
        return 0
    else:
```

```
        s = 0
        for sentence in article:
            token = sentence.split()
            n = len(token)
            s += n
        return s/len(article)
```

[5]:
```
def total_token(article):
    s = 0
    for sentence in article:
        token = sentence.split()
        n = len(token)
        s += n
    return s
```

[6]:
```
#input (list of dic)
#output numbers
def min_max_avg(jsons, name, type_no):
    Min = len(jsons[0][name])
    Max = len(jsons[0][name])
    SUM = 0
    numbers = []

    for i in range(len(jsons)):
        article = jsons[i][name]
        if type_no == 1:
            N = avg_token(article)
        elif type_no == 2:
            N = total_token(article)
        else:
            N = len(article)

        numbers.append(N)
        SUM += N
        if Min>N:
            Min = N
        if Max<N:
            Max = N

    Avg = SUM//len(jsons)

    return (Min, Max, Avg, len(jsons), numbers)
```

1.0.3 Print Output Functions

[7]:
```
from matplotlib import pyplot as plt

def plot_graph(bin_list, numbers, image_name):
    plt.hist(numbers, bins = bins_list)
    plt.savefig(image_name)
    plt.show()
```

[8]:
```
def print_result(title, Min, Max, Avg, l):
    #--------------------------------
    print(title)
    print('-------------------------------------------------------------')
    print('Number of articles:'+str(l))
    print('Longest:'+str(Max))
    print('Shortest:'+str(Min))
    print('Average:'+str(Avg))
```

[9]:
```
def print_out(jsons, name_str, output_str, type_no):
    Min, Max, Avg, l, numbers = min_max_avg(jsons, name_str, type_no)
    print_result(output_str, Min, Max, Avg, l)
    return numbers
```

2 Test data

[10]:
```
test = readfile('pubmed-dataset/test.txt')
test_jsons= list2json(test)
```

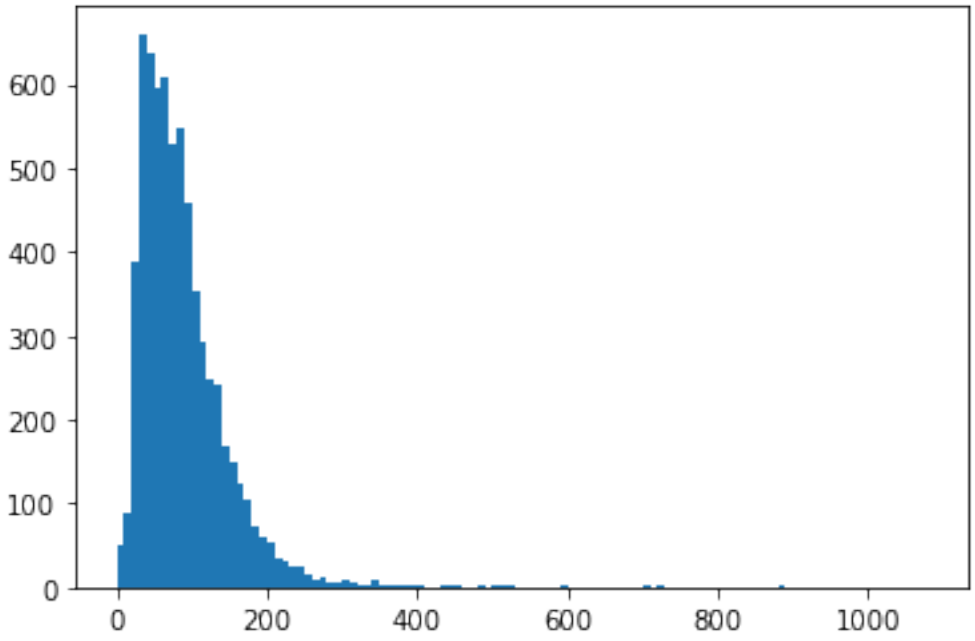
2.0.1 Test data: number of sentences in an article

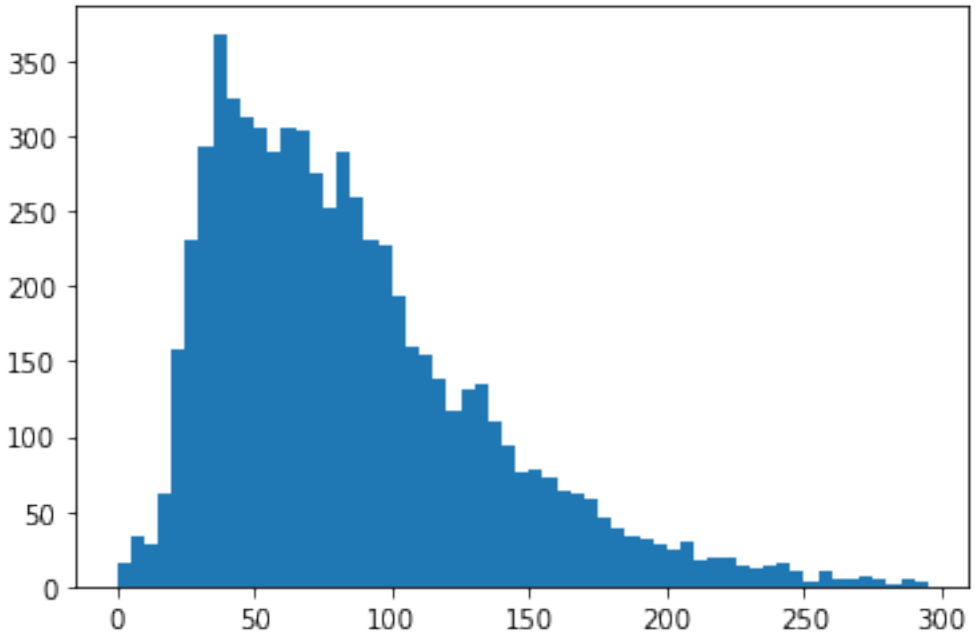
[11]:
```
test_s_numbers = print_out(test_jsons, 'article_text', 'Test Data', 3)
```

```
Test Data
-------------------------------------------------------------
Number of articles:6658
Longest:1081
Shortest:1
Average:87
```

[12]:
```
bins_list = list(range(0,1090,10))
plot_graph(bins_list, test_s_numbers, 'test_s_1.png')

bins_list = list(range(0,300,5))
plot_graph(bins_list, test_s_numbers, 'test_s_2.png')
```





2.0.2 Test data: number of tokens in a sentence

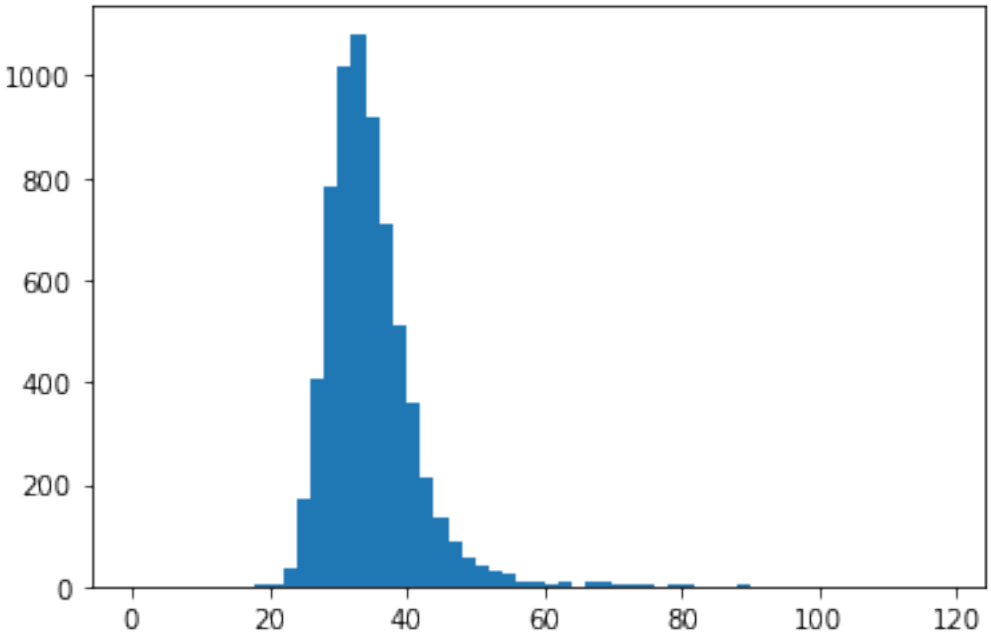
[13]:
```
test_t1_numbers = print_out(test_jsons, 'article_text', 'Test Data', 1)
```

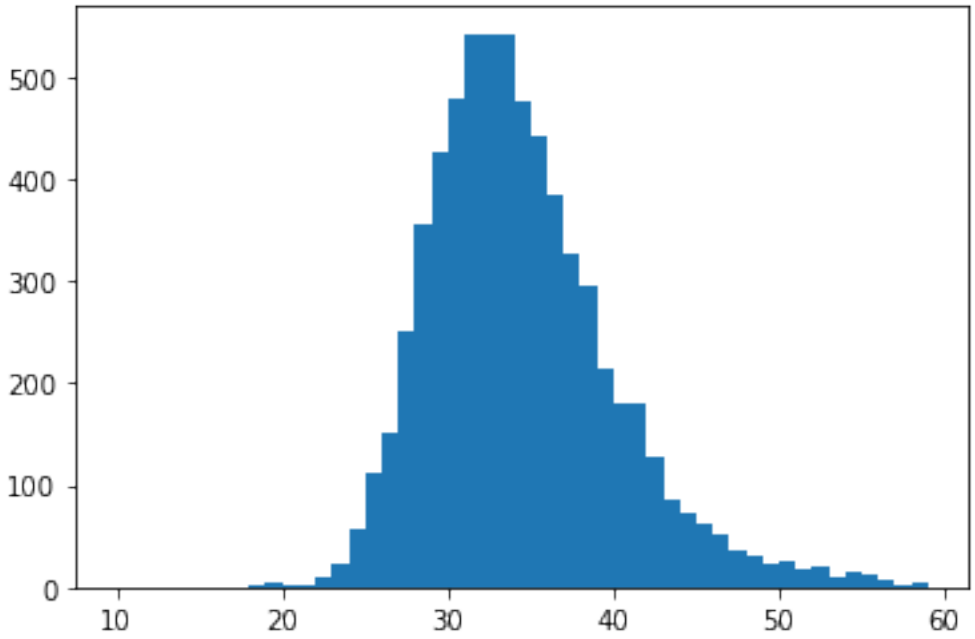
```
Test Data
```

```
------------------------------------------------------------
Number of articles:6658
Longest:114.68888888888888
Shortest:18.666666666666668
Average:34.0
```

[14]:
```
bins_list = list(range(0, 120, 2))
plot_graph(bins_list, test_t1_numbers, 'test_t1_1.png')

bins_list = list(range(10, 60, 1))
plot_graph(bins_list, test_t1_numbers, 'test_t1_2.png')
```





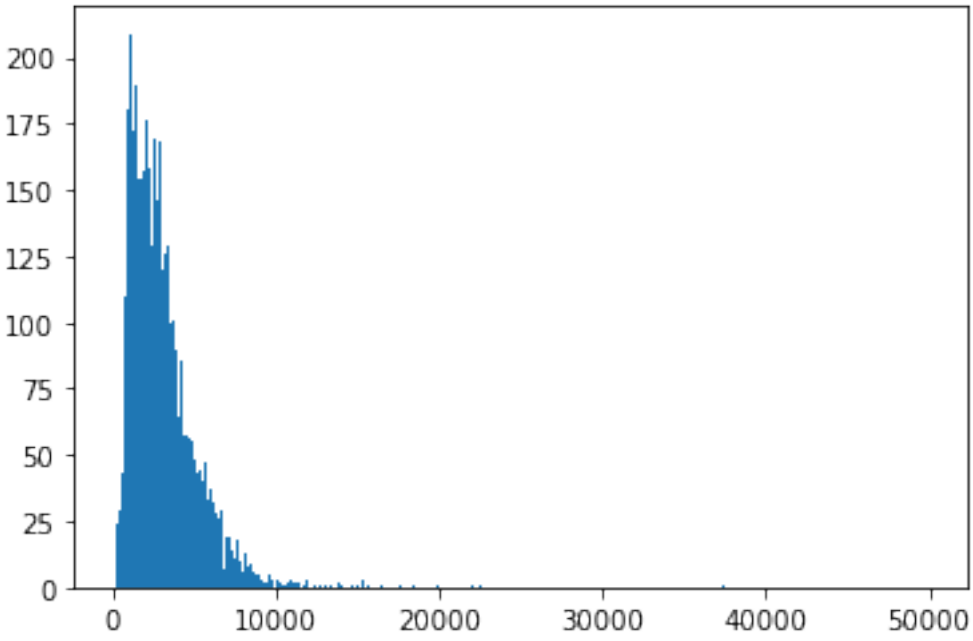
2.0.3 Test data: number of tokens in an article

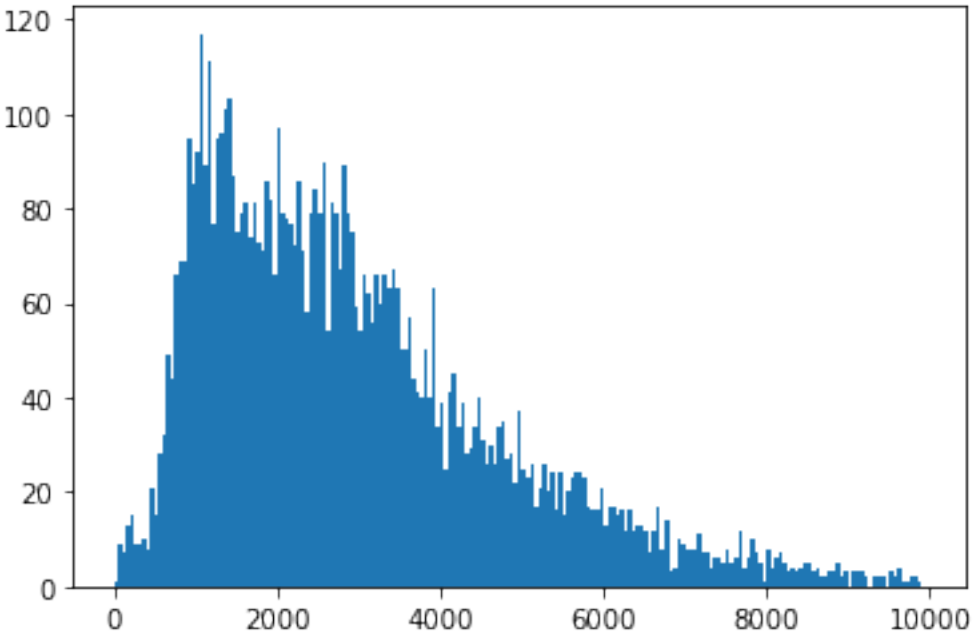
[15]:
```
test_t2_numbers = print_out(test_jsons, 'article_text', 'Test Data', 2)
```

```
Test Data
------------------------------------------------------------
Number of articles:6658
Longest:48750
Shortest:20
Average:3092
```

[16]:
```
bins_list = list(range(100,50000,100))
plot_graph(bins_list, test_t2_numbers, 'test_t2_1.png')

bins_list = list(range(0,10000,50))
plot_graph(bins_list, test_t2_numbers, 'test_t2_2.png')
```


200
175
150
125
100
75
50
25
0
0
10000
20000
30000
40000
50000


120
100
80
60
40
20
0
0
2000
4000
6000
8000
10000

3 Train data

[17]:
```
start = time.time()
#--------------------------------------------------------------------------
train = readfile('pubmed-dataset/train.txt')
train_jsons= list2json(train)
#--------------------------------------------------------------------------
print(time.time()-start)
```

```
62.17582893371582
```

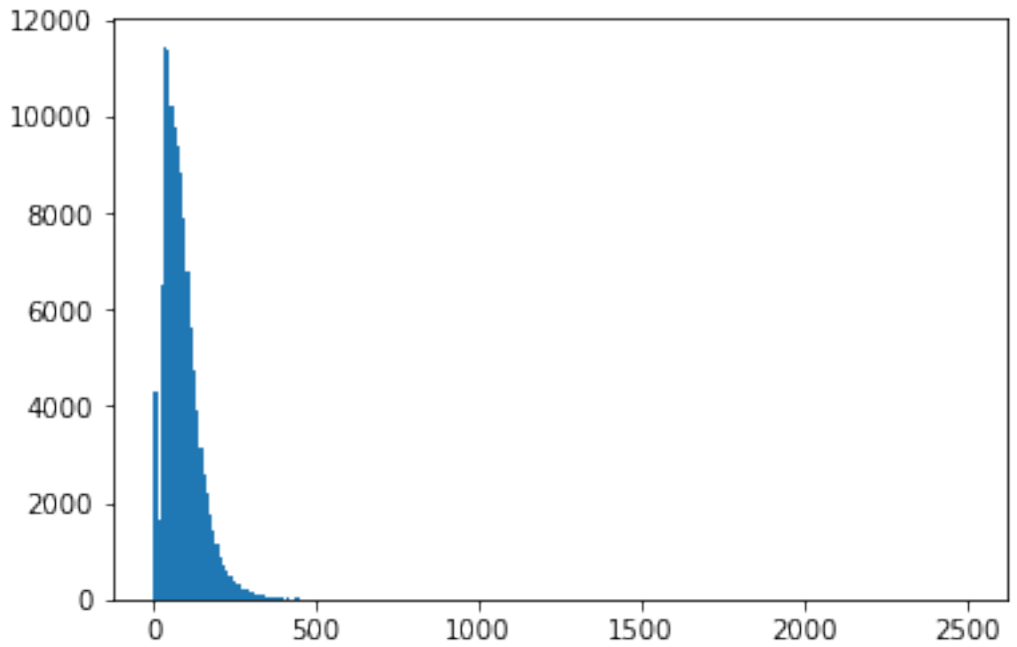
3.0.1 Train data: number of sentences in an article

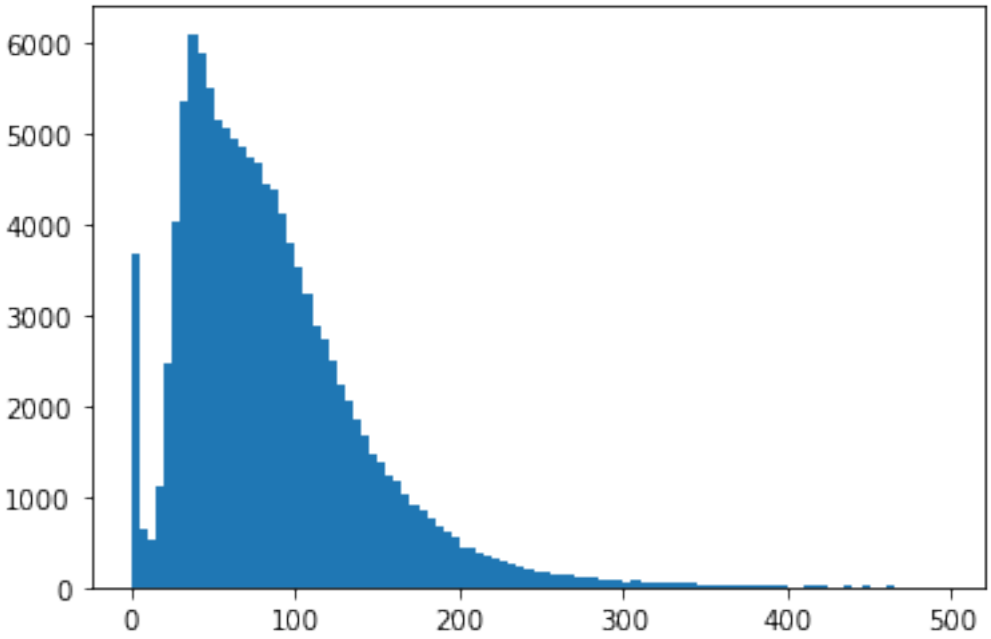
[18]:
```
train_s_numbers = print_out(train_jsons, 'article_text', 'Train Data', 3)
```

```
Train Data
------------------------------------------------------------
Number of articles:119924
Longest:2509
Shortest:0
Average:86
```

[19]:
```
bins_list = list(range(0,2510,10))
plot_graph(bins_list, train_s_numbers, 'train_s_1.png')

bins_list = list(range(0,500,5))
plot_graph(bins_list, train_s_numbers, 'train_s_2.png')
```



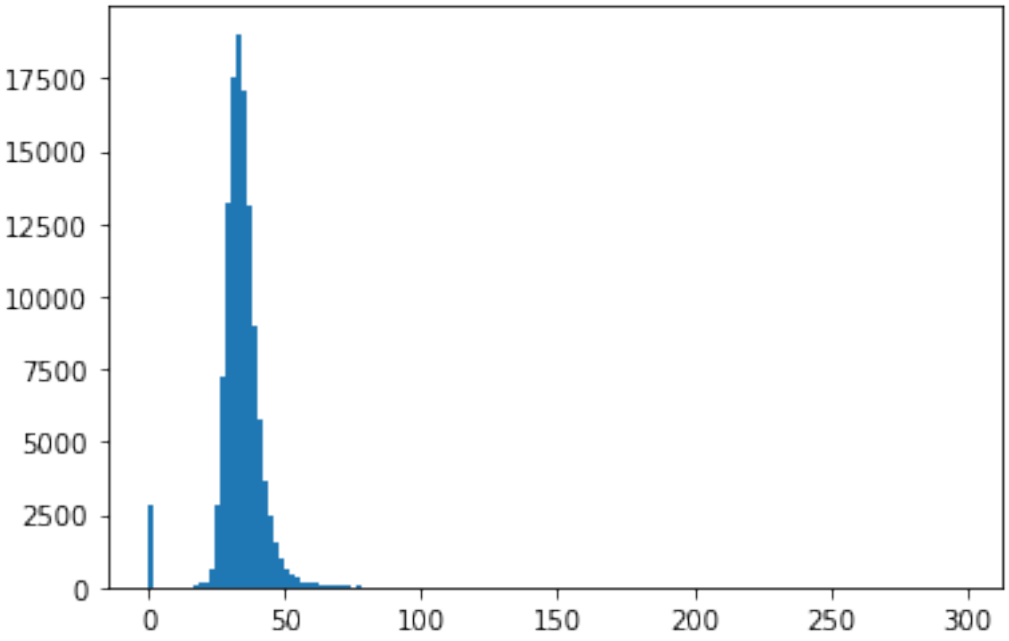


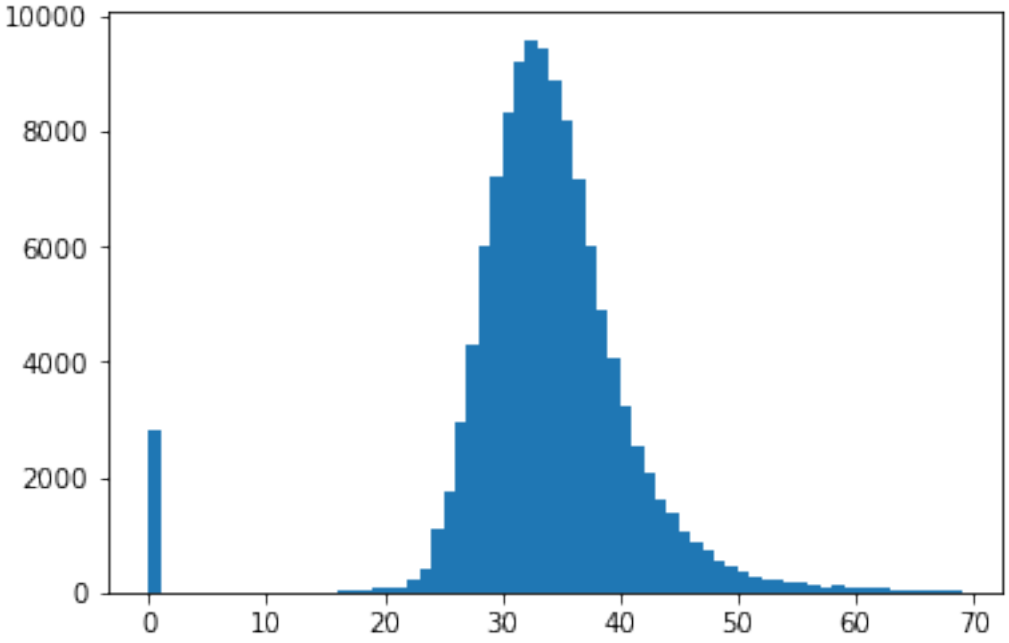
3.0.2 Train data: number of tokens in a sentence

[20]:
```
train_t1_numbers = print_out(train_jsons, 'article_text', 'Train Data', 1)
bins_list = list(range(0, 300, 2))
plot_graph(bins_list, train_t1_numbers, 'train_t1_1.png')

bins_list = list(range(0, 70, 1))
plot_graph(bins_list, train_t1_numbers, 'train_t1_2.png')
```

```
Train Data
------------------------------------------------------------
Number of articles:119924
Longest:275.1666666666667
Shortest:0.0
Average:33.0
```


17500
15000
12500
10000
7500
5000
2500
0
0
50
100
150
200
250
300


10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60
70

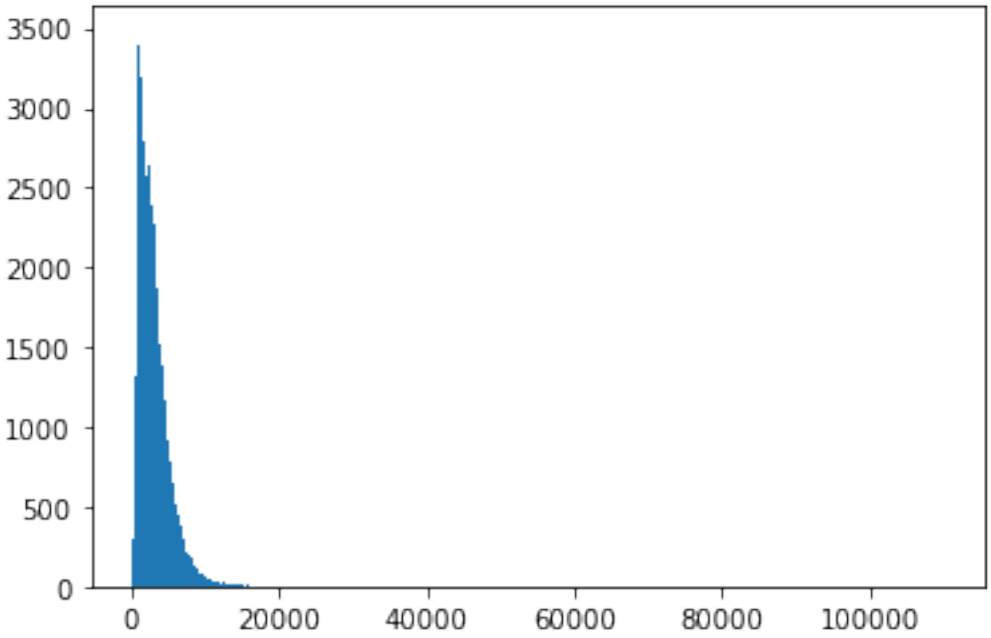
3.0.3 Train data: number of tokens in an article

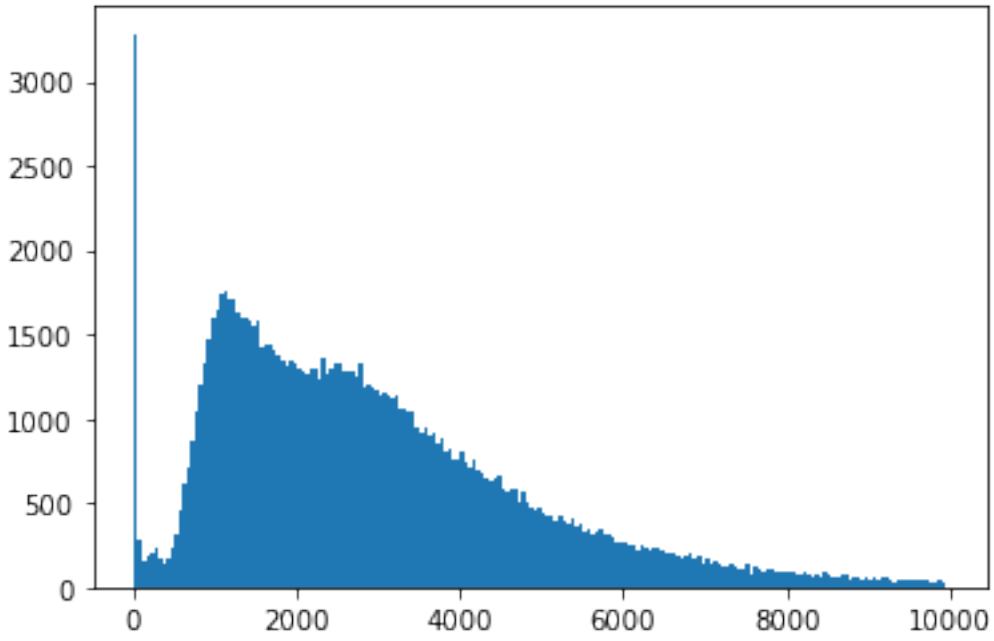
[21]:
```
train_t2_numbers = print_out(train_jsons, 'article_text', 'Train Data', 2)
```

```
Train Data
------------------------------------------------------------
Number of articles:119924
Longest:109759
Shortest:0
Average:3043
```

[22]:
```
bins_list = list(range(100,110000,100))
plot_graph(bins_list, train_t2_numbers, 'train_t2_1.png')

bins_list = list(range(0,10000,50))
plot_graph(bins_list, train_t2_numbers, 'train_t2_2.png')
```





3.0.4 Validation

[23]:
```
start = time.time()
#--------------------------------------------------------------------------
val = readfile('pubmed-dataset/val.txt')
val_jsons= list2json(val)
#--------------------------------------------------------------------------
print(time.time()-start)
```

```
1.5881285667419434
```

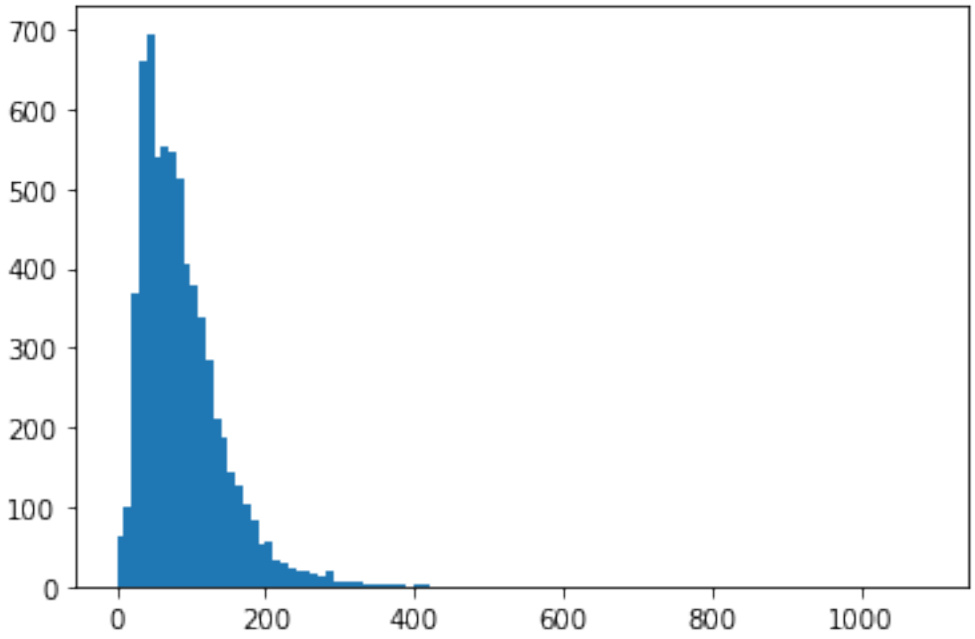
3.0.5 Validation data: number of sentences in an article

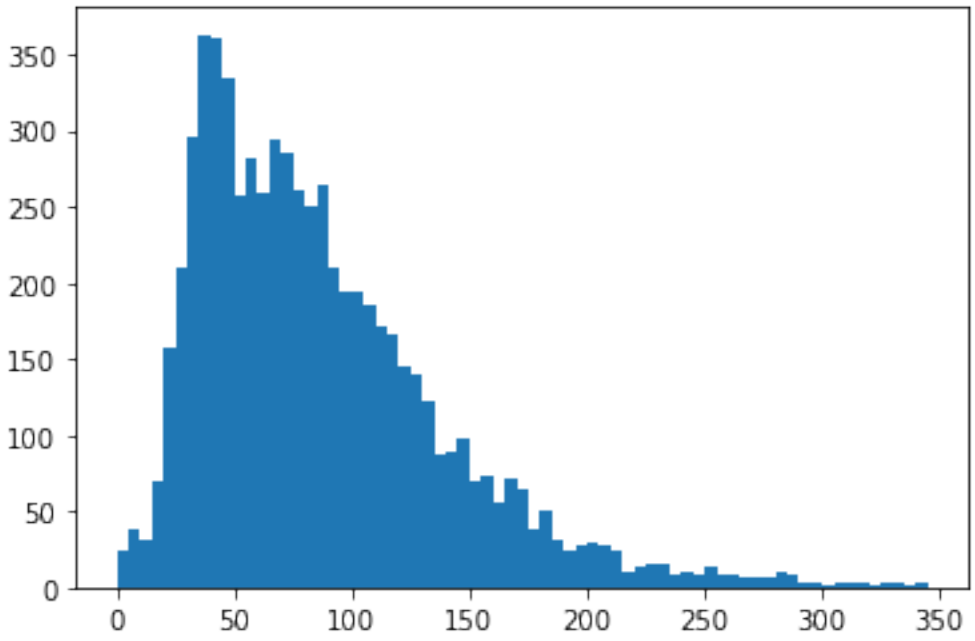
[24]:
```
print(len(val_jsons))
val_s_numbers = print_out(val_jsons, 'article_text', 'Validation Data', 3)
```

```
6633
Validation Data
-------------------------------------------------------------
Number of articles:6633
Longest:1085
Shortest:0
Average:87
```

[25]:
```
bins_list = list(range(0, 1100, 10))
plot_graph(bins_list, val_s_numbers, 'val_s_1.png')
```

```
bins_list = list(range(0, 350, 5))
plot_graph(bins_list, val_s_numbers, 'val_s_2.png')
```



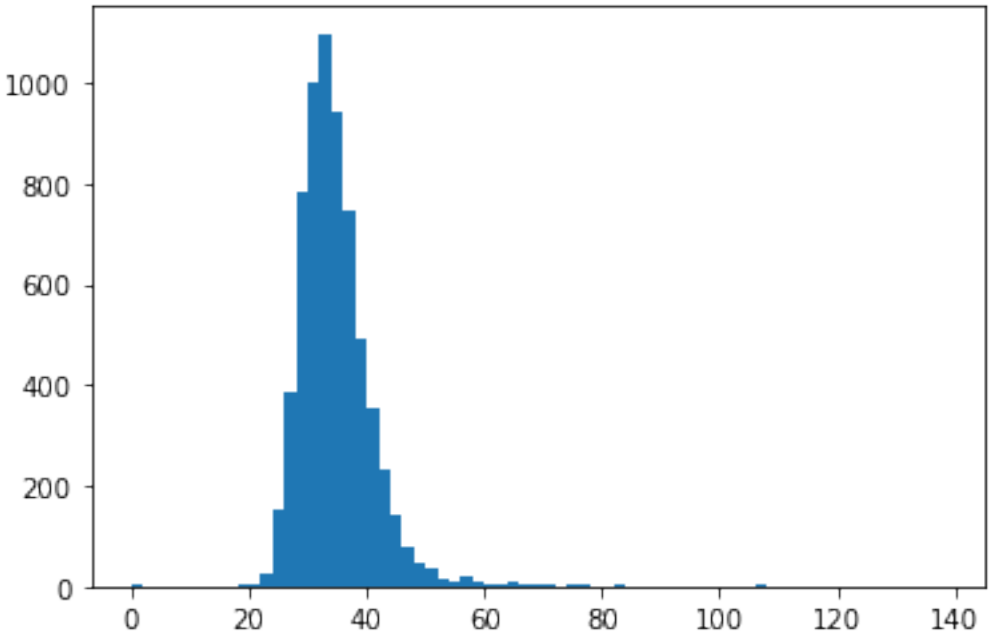


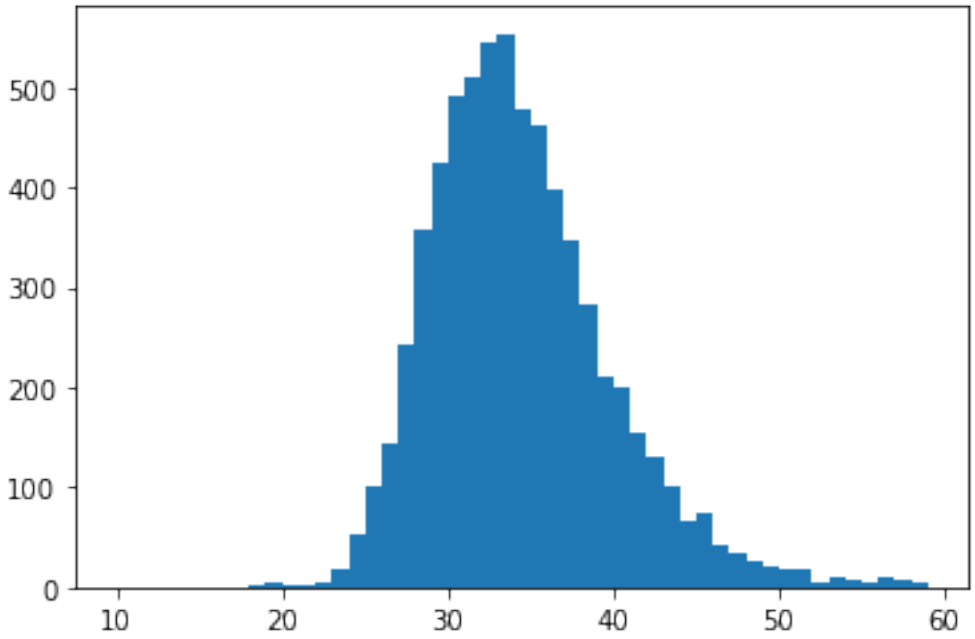
3.0.6 Validation data: number of tokens in a sentence

[26]:
```
val_t1_numbers = print_out(val_jsons, 'article_text', 'Validation Data', 1)
bins_list = list(range(0, 140, 2))
plot_graph(bins_list, val_t1_numbers, 'val_t1_1.png')

bins_list = list(range(10, 60, 1))
plot_graph(bins_list, val_t1_numbers, 'val_t1_2.png')
```

```
Validation Data
------------------------------------------------------------
Number of articles:6633
Longest:139.55555555555554
Shortest:0
Average:34.0
```



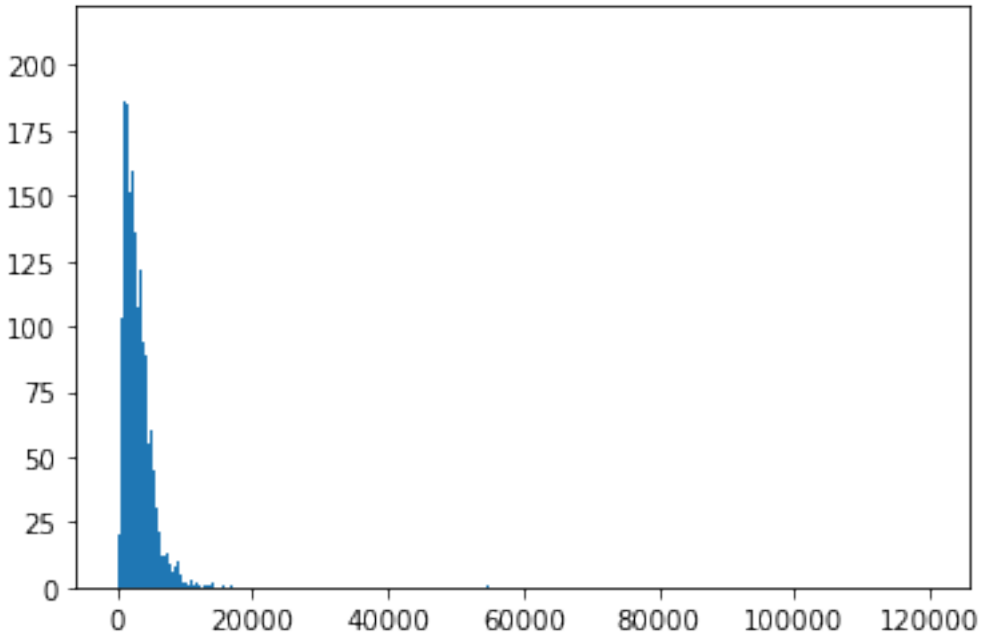


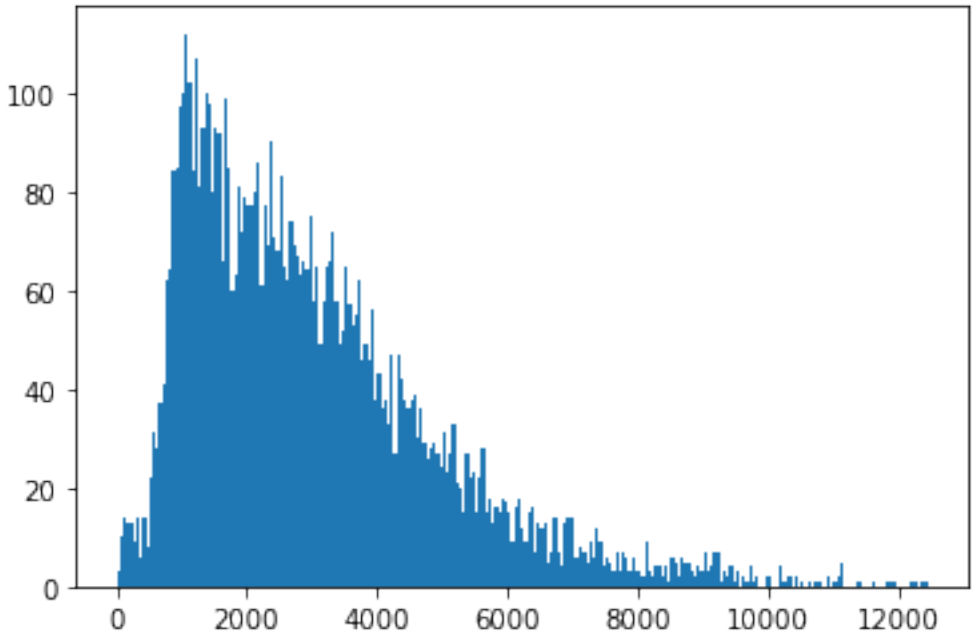
3.0.7 Validation data: number of tokens in an article

[27]:
```
val_t2_numbers = print_out(val_jsons, 'article_text', 'Validation Data', 2)
bins_list = list(range(0,120000,100))
plot_graph(bins_list, val_t2_numbers, 'val_t2_1.png')

bins_list = list(range(0,12500,50))
plot_graph(bins_list, val_t2_numbers, 'val_t2_2.png')
```

```
Validation Data
------------------------------------------------------------
Number of articles:6633
Longest:119269
Shortest:0
Average:3111
```


200
175
150
125
100
75
50
25
0
0
20000
40000
60000
80000
100000
120000


100
80
60
40
20
0
0
2000
4000
6000
8000
10000
12000

4 Train+Validation+Test

[28]:
```
all_jsons = train_jsons + val_jsons + test_jsons
```

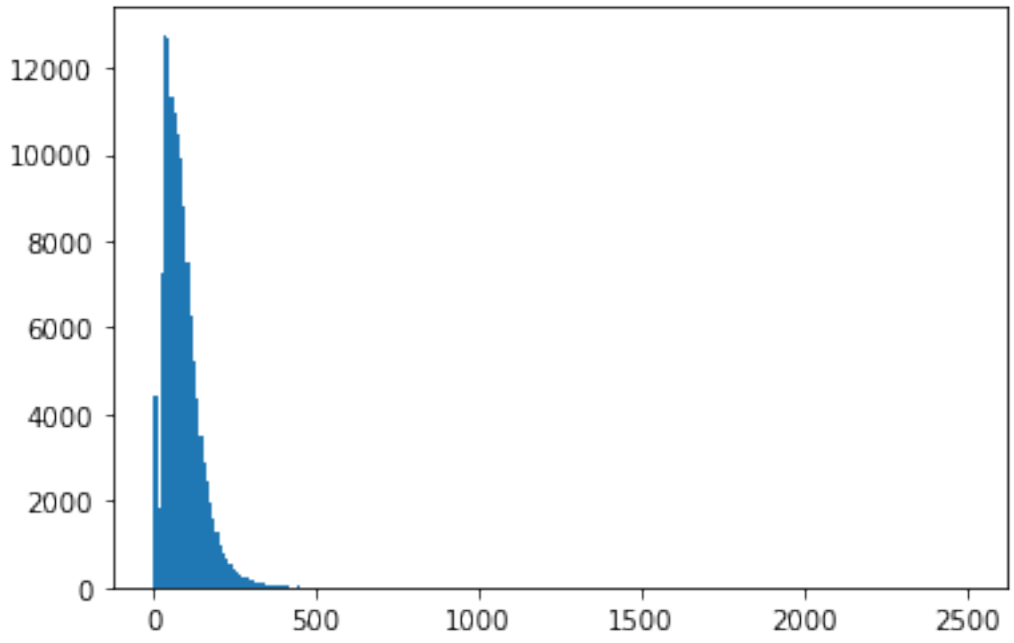
4.0.1 Number of sentences in an article

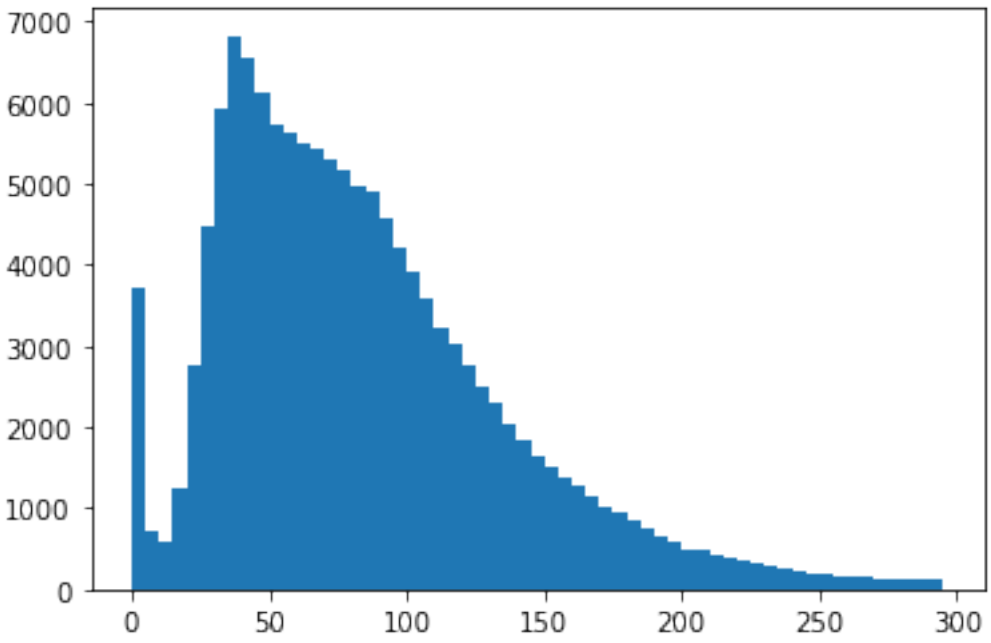
[29]:
```
s_numbers = print_out(all_jsons, 'article_text', 'Validation Data', 3)
```

```
Validation Data
------------------------------------------------------------
Number of articles:133215
Longest:2509
Shortest:0
Average:86
```

[30]:
```
bins_list = list(range(0,2510,10))
plot_graph(bins_list, s_numbers, 's_1.png')

bins_list = list(range(0,300,5))
plot_graph(bins_list, s_numbers, 's_2.png')
```



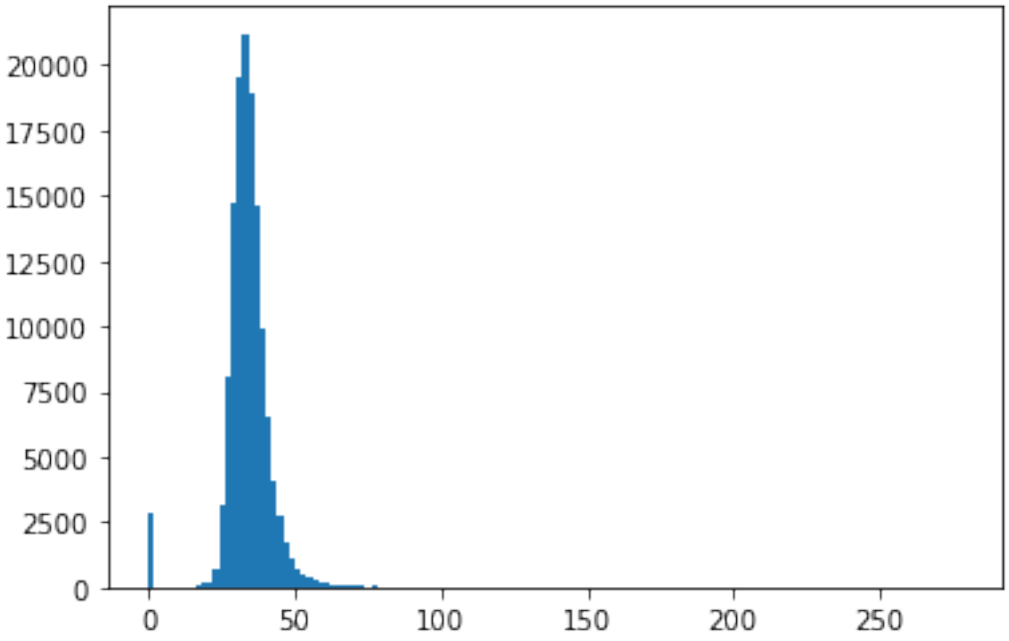


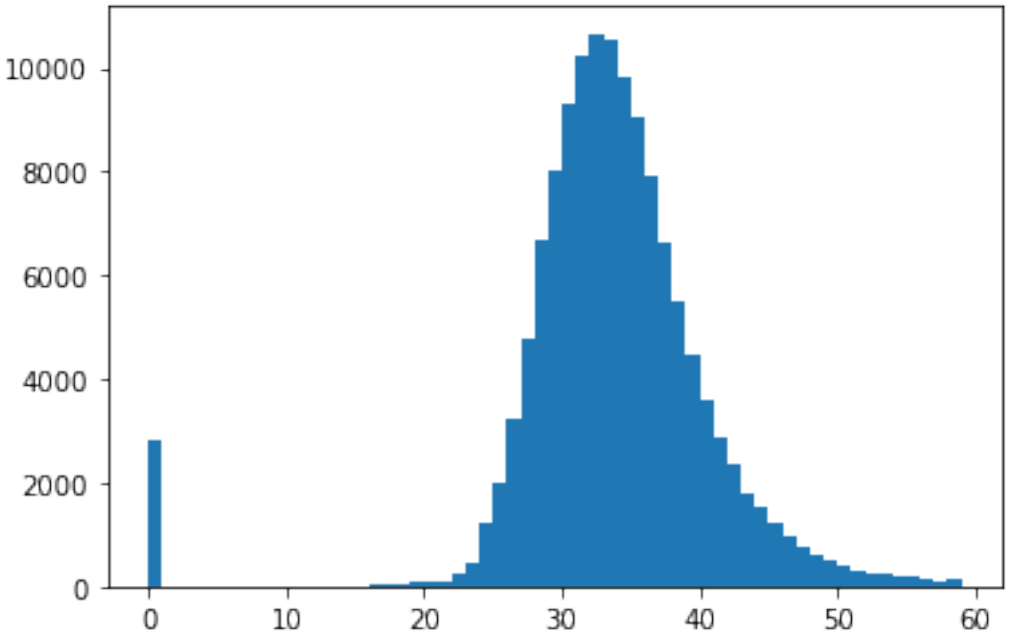
4.0.2 Number of tokens in a sentence

[31]:
```
t1_numbers = print_out(all_jsons, 'article_text', 'Validation Data', 1)
bins_list = list(range(0, 280, 2))
plot_graph(bins_list, t1_numbers, 't1_1.png')

bins_list = list(range(0, 60, 1))
plot_graph(bins_list, t1_numbers, 't1_2.png')
```

```
Validation Data
------------------------------------------------------------
Number of articles:133215
Longest:275.1666666666667
Shortest:0.0
Average:33.0
```


20000
17500
15000
12500
10000
7500
5000
2500
0
0
50
100
150
200
250

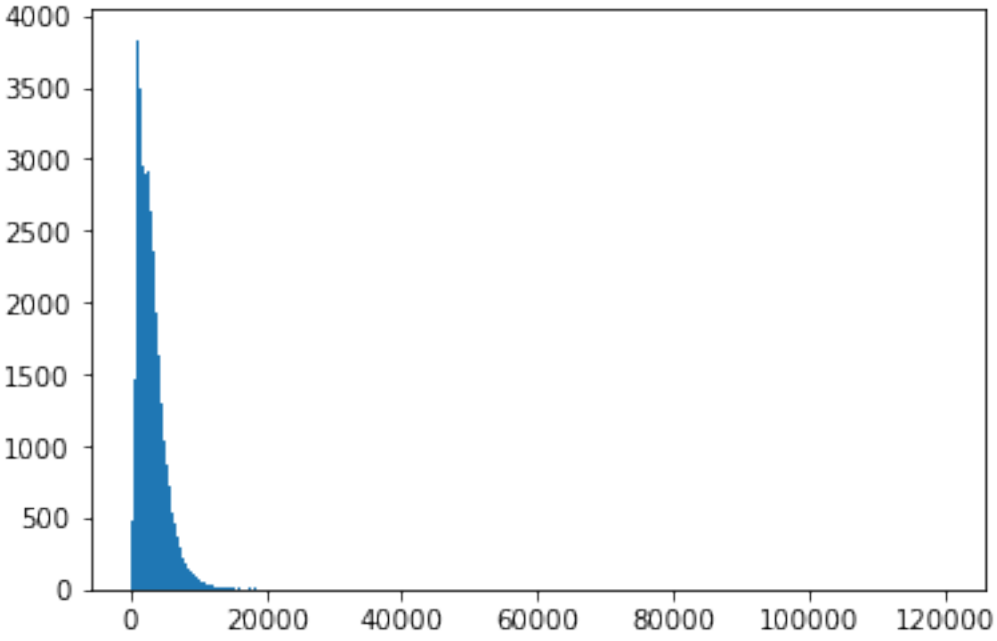

10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60

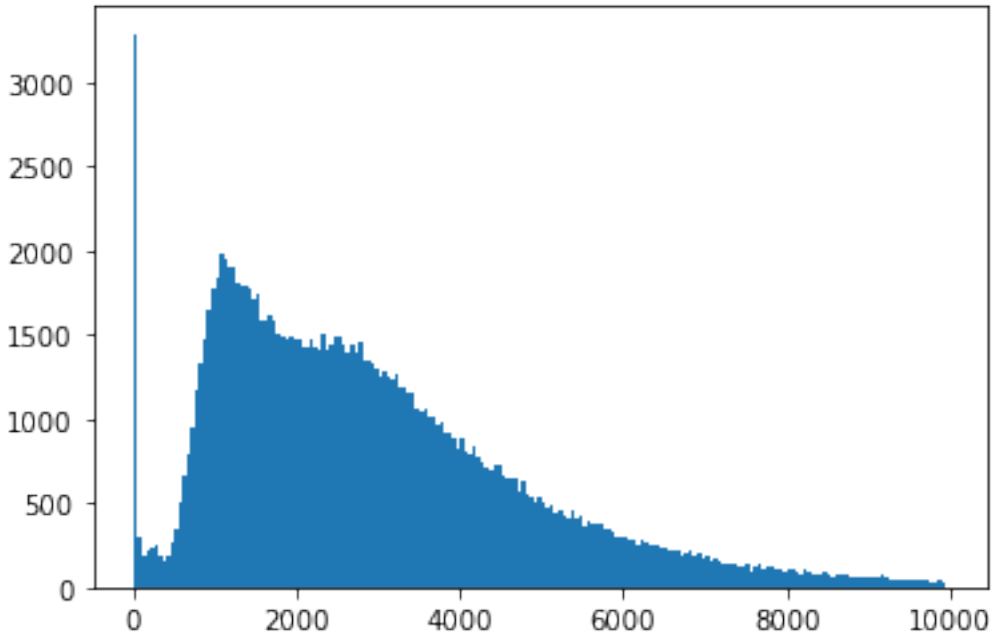
4.0.3 Number of tokens in an article

[32]:
```
t2_numbers = print_out(all_jsons, 'article_text', 'Validation Data', 2)
bins_list = list(range(0,120000,100))
plot_graph(bins_list, t2_numbers, 't2_1.png')

bins_list = list(range(0,10000,50))
plot_graph(bins_list, t2_numbers, 't2_2.png')
```

```
Validation Data
------------------------------------------------------------
Number of articles:133215
Longest:119269
Shortest:0
Average:3048
```




3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000